AMENDMENTS TO THE SPECIFICATION

A. Please replace the paragraph starting on page 7, line 4, with the following replacement paragraph (marked up to show changes made).

In Fig. 2A we face a more difficult challenge for the DCT basis functions. Now there is a very sharp transition or edge in the domain. Fig. 2B shows the discretized version of the continuous signal exhibited in Fig. 2A. Fig. 2C shows the reconstruction results based on maintaining a small number of coefficients. Now the residual error is very high. Fig. 2D depicts a case where many coefficients are used and the associated residual error is very small. It can be shown that the amount of error of the reconstruction is inversely proportional to the number of coefficients that are preserved and therefore must <u>be</u> transmitted to the decoder. Thus the DCT basis functions are not very efficient in this case. Note that the DCT is the primary transform of choice in the JPEG and MPEG 1-2-4 families of standards, though MPEG-4 allows for other transforms. In general the DCT does not perform well near sharp edges.

B. Please replace the paragraph starting on page 15, line 18, with the following replacement paragraph (marked up to show changes made).

In the presence of quantization the process is slightly more complicated. Remember that if high compression ratios are desired then having to send the exact representation of the error differences at each level will be very costly in terms of bits. To avoid this it is necessary to quantize the data in such a way that reasonable quality is achieved on the decoder side. [[Fig 9A-C]] <u>Fig. 9A-9B</u> depicts an example of quantization. In Fig. 9A an example of a set of quantization intervals and their representative values are depicted. In Fig 9A, for all the error differences, E, if their value lies between —Q < E <+Q the quantized result will be zero. For all E such that +Q < e < +2Q the quantized result would be +3/2 Q and so on. The result of applying the quantization function described in [[Fig 9a]]

Fig. 9A to a set of 2-D sampled input data (as seen in the top portion of Fig 9b) where Q = 5 is also displayed in [[Fig 9b]] Fig. 9B at the bottom of the page.

C. Please replace the paragraph starting on page 16, line 14, with the following replacement paragraph (marked up to show changes made).

Many of the differences present in modem multi-scale transforms involve different approaches to the problem of optimal quantization in order to obtain the best possible reconstruction for a given bit rate. In addition, many conventional sub-band encoders will also separate each level of the pyramid into multiple sub-bands through an application of low-pass (i.e. averaging) and high-pass (i.e. differencing or predict) filters. Then the corresponding inverse transform with quantization involves separate logic for the reconstruction of a given sub-band at each finer scale of the multi-resolution pyramid. However, the basic framework of the forward and inverse transform is much the same as described above~~ in sections A-C~~.

D. Please replace the paragraph starting on page 18, line 1, with the following replacement paragraph (marked up to show changes made).

The present invention relates to the efficient application of multi-scale transforms to arbitrary shaped domains of an N-dimensional image. The above procedure of padding or extension is suitable only for rectangular domains. For instance, the approach of using a 2-D symmetric extension is not feasible for ~~arbitrarily~~ arbitrary shapes as in such cases a true 2D symmetric extension cannot even be defined. In Fig. 11, an example of generalized non-rectangular domains in 2-D is shown. Such shaped domains are encountered whenever an image processor segments an image frame and in fact MPEG-4 supports arbitrarily shaped video object layers. In principle the entire domain of the signal itself may be arbitrarily shaped or on the other hand the signal domain may be partitioned into a collection of arbitrarily shaped regions.

E. Please replace the paragraph starting on page 18, line 20, with the following replacement paragraph (marked up to show changes made).

One scheme that has been proposed for coding functions on arbitrary shaped domains is coding for polygonal meshes (see [Berg *I*]). The domain is tessellated into a grid of regular polygons (for example triangles). The function is assumed to be well represented by its values at the polygonal vertices (termed nodes). These values are then differentially coded. Typically, the function values are linearly interpolated within the polygon. MPEG-4, for instance, supports coding of triangular 2-D meshes. The size of the polygons determines the accuracy of the coding. Large polygons produce few nodes and thus the coding is bit-efficient. The function is however very poorly approximated within large flat regions. If the polygons are small, the function is well approximated, but the large number of nodes results in very large bit costs for transmission.

F. Please replace the paragraph starting on page 23, line 6, with the following replacement paragraph (marked up to show changes made).

In another embodiment of the invention, pattern adaptive prediction is used when predicting the next finer level of the transform pyramid. The pattern adaptive prediction uses the parent grid to determine what geometry of a filter is to be used when predicting the child grid. At the boundaries of the domain, the pattern adaptive prediction can be coupled with the domain adaptive prediction technique.

G. Please replace the paragraph starting on page 24, line 9, with the following replacement paragraph (marked up to show changes made).

When an image segmentation routine based on *[Prakash II]* is applied to ~~Fig. 14B~~ Fig. 14A, the result in ~~Fig. 14C~~ Fig. 14B is obtained. In order to subdivide the image into meaningfully sized regions of roughly similar intensity values, the segmentation routine separated the scene into a multitude of arbitrarily shaped domains. Note that many (but not all) of the segment boundaries are spatially coincident (or very nearly so) with sharp edges in the input image. Also note that while some of the interiors of the segments will contain perceptible features, in general the pixel values internal to each segment, i.e. those not near or along a boundary, do not change significantly on short length scales. In the case where

the multi-scale transform is to be used as part of an exemplary encoder / decoder system, the presence of such a large number of arbitrary shaped domains with strong edges transitions at the boundaries will quickly erode the efficiency of the transform thereby producing inferior quality at a low bit rate.

H. Please replace the paragraph starting on page 29, line 8, with the following replacement paragraph (marked up to show changes made).

The inverse transform of a multi-scale transform is composed of one or more predicts (i.e. interpolating filter + sample shift) and / or inverse updates (i.e. convolution filter + expansion) for the generation of the next finer scale representations of the signal based <u>on</u> one or more, coarse scale representations. In an embodiment of the invention, the domain adaptive transform described above in the '**<u>Mechanics</u>**' sections is directly used so as to increase the performance of the inverse transform thereby increasing the efficiency of the multi-scale transform in the presence of arbitrarily shaped domains; irrespective of whether the coarser scale representations were also constructed with the use of the invention.[[.]] In another embodiment of the invention, the aforementioned domain adaptive transform is used in the application of a series of inverse update and prediction filters to a set of arbitrarily shaped segments in order to construct a mathematically efficient sub-band decomposition for the inverse portion of a multi-scale transform with sub-bands. In yet another embodiment of the invention, the domain adaptive transform is applied in both the forward and inverse transforms of a multi-scale transform for maximal efficiency.

I. Please replace the paragraph starting on page 30, line 17, with the following replacement paragraph (marked up to show changes made).

A pattern adaptive transform is a transform that adapts itself to the patterns inherently present in the data the transform being applied to. In particular, one embodiment of multi-scale pattern adaptive transforms will be described here. As was mentioned earlier, in a multi-scale pyramidal transform, the forward transform build coarser and coarser averages of the image data are produced. On the inverse transform, the coarser parent data are used to predict the data on the finer(child) scale. The data is not interpolated with constant filter

coefficients, rather the coefficients are scaled in the data dependent way. The interpolation filter thus adapts itself to the pattern of the data. Specifically in the current embodiment, a 4x4 set of parents is chosen for interpolating the child grid. Each point initially has a fixed coefficient of a 4x4 filter associated with it. The approximate gradient value to each of the 16 parent values from the center is then computed. Each of the filter coefficients is then scaled by the inverse of the gradient value. The new filter is re-normalized and then applied to interpolate the data. In Fig. 17A, [[we]] an example of a "diagonal trough". The low lying line of the "trough" going from lower left to upper right are the low points emphasized in gray. In case of the "trough", the gradient values along the trough are small, while in the direction perpendicular to the trough are high. Thus the point in the middle will <u>be</u> interpolated primarily along the "equipotential" lines roughly parallel to die "trough", with the weight of the other points being quite small. The "trough" can also have a bend as illustrated by Fig. 17B. Here, the low lying line of the "trough" is again emphasized in gray, but it is no longer straight. The interpolation will still happen along the "equipotential" lines, this time approximately following the curve of the "trough". Fig 17C contains a "slanted surface". For the slanted surface, the low lying line of the trough is again emphasized in gray, here going from bottom left to top left. Again, the interpolation will happen mainly along the constant contour lines (up/down in this case) of the slanted surface. Note that no edge detection needs to be performed.

J. Please replace the paragraph starting on page 31, line 14, with the following replacement paragraph (marked up to show changes made).

______In another embodiment, the pattern adaptive transform is combined with domain adaptive transform to efficiently predict data near boundaries of domains. The filter coefficients for the interior points is first by the domain adaptive technique which redistributes the weight of the coefficients corresponding to external points. Then, the ~~patern~~ <u>pattern</u> adaptive technique is used to scale a renormalize [[thos]] <u>those</u> weights according to the pattern adaptive technique.